IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of:
Bryan M. Cantrill

Confirmation No.: 2597

Application No.: 10/699,062

Art Unit: 2162

Filed: October 31, 2003

Examiner: D. Y. Myint

For: MECHANISM FOR DATA AGGREGATION IN A TRACING FRAMEWORK

REVISED DECLARATION UNDER TO 37 C.F.R. § 1.131

In connection with Applicant's Response to the final Office Action issued on November 2, 2006, this declaration sets forth the pertinent facts proving conception and actual reduction to practice of the claimed invention prior to **May 16, 2002**.

1. I, Bryan Cantrill, am the sole inventor listed on U.S. Patent Application Serial No. 10/699,062 entitled "MECHANISM FOR DATA AGGREGATION IN A TRACING FRAMEWORK" filed on October 31, 2003.
2. I conceived and completed the actual reduction to practice of the claimed invention at least prior to March 12, 2002, when I gave an internal company speech directed, in part, to the claimed invention.
3. The speech, which was conducted on March 12, 2002, included a slide presentation and a live demonstration of the claimed invention. A copy of relevant portions of the slide presentation entitled "DTrace: Dynamic Tracing For Solaris" dated March 11, 2002, and a DVD video of the speech showing the live presentation and demonstration dated March 12, 2002, were previously

submitted in the Declaration Under 37 C.F.R. § 1.131 filed with the USPTO on September 5, 2006.

4. The portion of the DVD Video particularly related the subject matter of the referenced application is: 2:20:49 – 02:36:10. The aforementioned times are listed in the following format HH:MM:SS.

5. A concise mapping of the claims to the slide presentation and the DVD video is included under Tab 1.

6. All events related to the conception and completion of the actual reduction to practice of the claimed invention were performed in the United States.

I, Bryan M. Cantrill, hereby declare that all statements made herein of my own knowledge are true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

Signed this day 2, of February 2007

Bryan M. Cantrill

TAB 1

MAPPING OF PENDING CLAIMS TO SLIDE PRESENTATION AND DVD VIDEO

The following mapping should not be constructed to mean that support is only provided in the portions of the Slide Presentation and DVD Video listed.

CLAIM NO.	LIMITATION	SLIDE PRESENTATION	DVD VIDEO*
1	obtaining data from the kernel; and	73	2:28:40 2:30:00
	storing the data in a data set in an aggregation buffer using an aggregation function.	68 71 72	2:23:20 2:27:40 2:28:05 2:30:00
2	wherein the data set comprises a key component, an aggregation identifier component, and a value component.	72	2:28:05
3	wherein obtaining data comprises: obtaining an expression, a new value, and an aggregation identifier, and	68	2:23:20
	generating a key using the expression and the aggregation identifier.	72	2:28:05 2:30:00
4	wherein storing the data set comprises:		

* All times listed under "DVD Video" correspond to the starting time of a portion of the DVD Video, which includes the limitation in question. Further, all times are listed in the following format: HH:MM:SS.

CLAIM NO.	LIMITATION	SLIDE PRESENTATION	DVD VIDEO*
	storing the key in the key component,	72	2:28:05 2:30:00
	storing the aggregation identifier in the aggregation identifier component, and	72	2:28:05 2:30:00
	updating a current value in the value component using the new value and the aggregation function.	72	2:28:05 2:30:00
5	wherein storing the data set further comprises: generating a hash key using the key;	72	2:28:05
	searching for a hash bucket corresponding to the key using the hash key;	72	2:28:05 2:30:00
	searching for a hash chain element in the hash bucket corresponding to the key;	72	2:28:05 2:30:00
	updating the value component of the data set associated with the hash chain element if a hash chain element corresponding to the key is found, wherein the updating the value component comprises applying the aggregation function to the current value in the value component using the new value as input;	72	2:28:05 2:30:00
	creating a new hash chain element if the hash chain element corresponding to the key is not found, wherein creating a new hash chain element comprises associating a new data set with the new hash chain element, storing the key in a key component of the new data set, storing the aggregation identifier in an aggregation identifier component of the new data set, and storing an initial value in a value component of the new data set; and	72	2:28:05 2:30:00

CLAIM NO.	LIMITATION	SLIDE PRESENTATION	DVD VIDEO*
	updating the value component associated with the new hash chain element, wherein the updating the value component associated with the new hash chain element comprises applying the aggregation function to the initial value using the new value as input.	72	2:28:05 2:30:00
6	wherein the data set comprises a key component, an aggregation identifier component, and a value component, comprising: obtaining an expression, a new value, and an aggregation identifier;	68 72	2:23:20 2:28:05
	generating a key using the expression and the aggregation identifier; and	72	2:28:05 2:30:00
	storing the data set in a buffer, wherein storing the data set comprises storing the key in the key component, storing the aggregation identifier in the aggregation identifier component, and updating a current value in the value component using the new value and an aggregation function.	72	2:28:05 2:30:00
7	wherein storing the data set comprises: generating a hash key using the key;	72	2:28:05
	searching for a hash bucket corresponding to the key using the hash key;	72	2:28:05 2:30:00
	searching for a hash chain element in the hash bucket corresponding to the key;	72	2:28:05 2:30:00
	updating the value component of the data set associated with the hash chain element if a hash chain element corresponding to the key is found, wherein the updating the value component comprises applying the aggregation function to the current value in the value component using the new value as input;	72	2:28:05 2:30:00

CLAIM NO.	LIMITATION	SLIDE PRESENTATION	DVD VIDEO*
	creating a new hash chain element if the hash chain element corresponding to the key is not found, wherein creating a new hash chain element comprises associating a new data set with the new hash chain element, storing the key in a key component of the new data set, storing an aggregation identifier in the aggregation identifier component of the new data set, and storing an initial value in a value component of the new data set; and	72	2:28:05 2:30:00
	updating the value component associated with the new hash chain element, wherein the updating the value component associated with the new hash chain element comprises applying the aggregation function to the initial value using the new value as input.	72	2:28:05 2:30:00
8	wherein the hash chain element is associated with the data set using a pointer.	72	2:28:05 2:30:00
9	wherein the new hash chain element is associated with the new data set using a pointer.	72	2:28:05 2:30:00
10	wherein the expression comprises an n-tuple.	68	2:23:20
11	Cancelled		
12	A method for integrating data into a user-level table, comprising:		
	obtaining a data set from an aggregation buffer, wherein the data set comprises a key component, an aggregation identifier component, and a value component;	73	2:28:40 2:30:00
	obtaining an aggregation identifier matching a value of the aggregation identifier in the aggregation identifier component of the data set to obtain a user-level table key;	72	2:28:05 2:30:00
	hashing the user-level table key to obtain a generated hash key;	72	2:28:05

CLAIM NO.	LIMITATION	SLIDE PRESENTATION	DVD VIDEO*
	Updating a value component of a user-level table entry if a user-level table entry having a hash key matching the generated hash key is found, wherein updating the value component comprises applying an aggregation function corresponding to the aggregation identifier to the value component using the new value as input;	72	2:28:05 2:30:00
	creating a new user-level table entry if a user-level table entry having a hash key matching the generated hash key is not found, wherein creating the new user-level table entry comprises storing the generated hash value, and an initial value in a value component of the new user-level table entry in the new user-level table entry; and	72	2:28:05 2:30:00
	updating the value component in the new user-level table entry, wherein updating the value component in the new user-level table entry comprises applying the aggregation function corresponding to the aggregation identifier to the value component in the new user-level table entry using the value component of the data set as input.	72	2:28:05 2:30:00
13	wherein the user-level table is a hash table.	72 73	2:28:05 2:28:40
14	wherein obtaining the aggregation identifier matching the value of the aggregation identifier comprises searching at least one selected from the group consisting of a user-level dictionary and a kernel level dictionary.	72 73	2:28:05 2:28:40
15	A data aggregation buffer comprising:	72	2:28:05
	a data set, wherein the data set comprises a key component for storing a key, an aggregation identifier component for storing an aggregation identifier component, and a value component for storing a value;	72	2:28:05 2:30:00
	a hash chain referencing the key component of the data set using a pointer; and	72	2:28:05 2:30:00

CLAIM NO.	LIMITATION	SLIDE PRESENTATION	DVD VIDEO*
	a hash bucket organizing the hash chain.	72	2:28:05 2:30:00
16	wherein the key is generated from a function using an aggregation identifier.	72	2:28:05 2:30:00
17	wherein the key is generated from a function using an aggregation identifier and an expression.	72	2:28:05 2:30:00
18	Cancelled		
19	A data aggregation system comprising:	72	2:28:05
	an aggregation buffer associated with a central processing unit;	72	2:28:05
	a user-level buffer operatively connected to the aggregation buffer and arranged to copy a data set from the aggregation buffer; and	72 73	2:28:05 2:28:40
	a user-level hash table storing a portion of the data set from the user-level buffer using a user-level dictionary,	72 73	2:28:05 2:28:40
	wherein the user-level dictionary provides a reference into the user-level hash table.	72 73	2:28:05 2:28:40
20	wherein the data aggregation buffer comprises:	72	2:28:05 2:30:00
	a data set, wherein the data set comprises a key component for storing a key, an aggregation identifier component for storing an aggregation identifier component, and a value component for storing a value;	72	2:28:05 2:30:00
	a hash chain referencing the key component of the data set using a pointer; and	72	2:28:05 2:30:00
	a hash bucket organizing the hash chain.	72	2:28:05 2:30:00

CLAIM NO.	LIMITATION	SLIDE PRESENTATION	DVD VIDEO*
21	wherein the key is generated from a function using an aggregation identifier.	72	2:28:05 2:30:00
22	wherein the key is generated from a function using an aggregation identifier and an expression.	72	2:28:05 2:30:00
23	An apparatus for obtaining data from a kernel, comprising:	73	2:28:40 2:30:00
	means for obtaining data from a kernel;		
	a tracing framework; and	67 68	2:22:15 2:23:20
	means for storing the data in a data set in the tracing framework,	68 72	2:23:20 2:28:05 2:30:00
	wherein the tracing framework comprises:	71	2:27:40
	An aggregation buffer associated with a central processing unit;	72	2:28:05
	a user-level buffer operatively connected to the aggregation buffer copying the data set from the aggregation buffer; and	73	2:28:40 2:30:00
	a user-level hash table storing a portion of the data set from the user-level buffer using a user-level dictionary,	72	2:28:05 2:30:00
	wherein the user-level dictionary provides a reference into the user-level hash table.	72	2:28:05 2:30:00
24	A computer system on a network obtaining data from a kernel comprising:	67	2:22:15
	a processor;		
	a memory;	67	2:22:15
	a storage device; and	67	2:22:15

CLAIM NO.	LIMITATION	SLIDE PRESENTATION	DVD VIDEO*
	software instructions stored in the memory for enabling the computer system to:	68 71	2:23:20 2:27:40
	obtain data from the kernel; and	72	2:27:40
	store the data in a data set an aggregation buffer using an aggregation function.	73	2:28:40 2:30:00